REMARKS

The Examiner rejected claims 1-20 under 35 U.S.C. §102(e) as allegedly being anticipated by Mauricio Bretemitz, Jr. et al. (U.S. Patent No. 6,381,739).

Applicants respectfully traverse the §102 rejections with the following arguments.

35 U.S.C. §102(e)

The Examiner rejected claims 1-20 under 35 U.S.C. §102(e) as allegedly being anticipated by Mauricio Breternitz, Jr. et al. (U.S. Patent No. 6,381,739).

For clarification, Applicant notes that the description of prior art in the BACKGROUND OF THE INVENTION section (i.e., col. 1, line 25 - col. 4, line 43) in Breternitz is denoted herein as "Breternitz Prior Art", whereas the description of Breternitz' invention in the DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT and Claims sections (i.e., col. 6, line 14 - col. 40, line 10) in Breternitz is denoted herein as "Breternitz Invention".

Applicants respectfully contend that Breternitz Prior Art and Breternitz Invention are distinct and different references, which cannot be combined in a rejection under 35 U.S.C. §102. A rejection of a claim under 35 U.S.C. §102 must be based on a single reference and cannot be based on a combination of references.

Claims 1-6

The Examiner rejected claims 1-6 under 35 U.S.C. §102(e) as allegedly being anticipated by Breternitz Prior Art.

Applicants respectfully contend that Breternitz Prior Art does not anticipate claim 1, because Breternitz Prior Art does not teach each and every feature of claim 1.

As a first example of why Breternitz Prior Art does not anticipate claim 1, Breternitz Prior Art does not teach the feature: "computer software that includes one or more call statements and a procedure which is callable by the or each call statement and which has two or more code branches and control flow code for directing program flow to the code branches".

The Examiner argues that Breternitz, col. 1, line 26 and col. 2, line 4 teaches the preceding feature of claim 1. In response, Applicants respectfully contend that Breternitz, col. 1, line 26 and col. 2, line 4 does not disclose a call statement and does not disclose call parameters, and therefore does not teach the preceding feature of claim 1.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all

regarding "Breternitz does not anticipates a call statement and does not disclose call parameter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been defined as calling a procedure function.... Breternitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Breternitz, Col. 2, Line 40- 60 ...".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Breternitz Invention is specific to FIG. 14 of Breternitz Invention and does not apply to the rejection under Breternitz Prior Art. In particular, col. 1, lines 25-40 of Breternitz Prior Art makes it clear that the control flow graph of FIG. 1 of Breternitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Breternitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Breternitz Prior Art are not disclosed as representing subroutines or function calls in Breternitz Prior Art. The Examiner cannot combine Breternitz Invention with Breternitz Prior Art under 35 U.S.C. §102.

As a second example of why Breternitz Prior Art does not anticipate claim 1, Breternitz Prior Art does not teach the feature: "analysing the procedure to identify said control flow code and said code branches".

The Examiner argues that Breternitz Prior Art, col.. 2, lines 4-6 teaches the preceding feature of claim 1.

In response, Applicants respectfully contend that Breternitz Prior Art, col.. 1, lines 4-6 does not disclose analysing the procedure to "identify said control flow code and said code branches", as required by claim 1. Instead, Breternitz Prior Art, col.. 2, lines 4-6 discloses analysing the procedure to "determine efficient rearrangement of computer basic blocks in memory so that software executes in an efficient manner." Applicants maintain that Breternitz Prior Art, col. 2, line 4-6 is totally irrelevant to the preceding feature of claim 1.

As a third example of why Breternitz Prior Art does not anticipate claim 1, Breternitz Prior Art does not teach the feature: "identifying for each said code branch a new procedure

containing the respective code branch".

The Examiner argues that Breternitz Prior Art, col. 2, lines 10-13 teaches the preceding feature of claim 1.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, line 10-13, does not identify a new procedure, as required by claim 1. Breternitz Prior Art, col. 2, line 10-13 merely discusses a trace data file the records time-sequential execution flow of the computer program.. Applicants maintain that Breternitz Prior Art, col. 2, line 10-13 is totally irrelevant to the preceding feature of claim 1.

As a fourth example of why Breternitz Prior Art does not anticipate claim 1, Breternitz Prior Art does not teach the feature: "recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch".

The Examiner argues that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 teaches the preceding feature of claim 1.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 does not disclose the preceding feature of claim 1. Breternitz Prior Art, col. 2, lines 14-15 merely discloses storing block execution order in a time sequential manner. Breternitz Prior Art, col. 2, lines 41-45 merely discloses scanning trace data and analyzing pairs of basic blocks that are adjacent to each other in time. Breternitz Prior Art, col. 2, lines 41-45 does not discloses a list, and most certainly not disclose a list of data entries corresponding to the respective new procedures. Applicants maintain that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 is totally irrelevant to the preceding feature of claim 1.

As a fifth example of why Breternitz Prior Art does not anticipate claim 1, Breternitz Prior Art does not teach the feature: "for the or each call statement, scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters directed to said control flow code by the call statement and modifying the call

statement to replace the call to the original procedure by a call to the corresponding new procedure".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 teaches the preceding feature of claim 1.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the preceding feature of claim 1. Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the "scanning the entries ..." and the "modifying the call statement ..." aspects of the preceding feature of claim 1.

In further response, Applicants maintain that Bretemitz Prior Art does not teach the "call parameters" in the preceding feature of claim 7. In "Response to Arguments", the Examiner argues that "Bretemitz anticipates ... "call parameters" [1310, Fig. 31]". In response, Applicants maintain that the Examiner's citation of block 1310 in FIG. 31 of Bretemitz Invention does not apply to the rejection under Bretemitz Prior Art, since the Examiner cannot combine Bretemitz Invention with Bretemitz Prior Art under 35 U.S.C. §102. Moreover, the parameters a, dfs, p, n, chain in block 1310 in FIG. 31 of Bretemitz are not call parameters passed between subroutines or procedures as defined in Applicants' specification (and in the art of computer programming as well), but rather are variables that are initialized in block 1310 (see Bretemitz Invention, col. 23, lines 53-58).

Based on the preceding arguments, Applicants respectfully maintain that Bretemitz Prior Art does not anticipate claim 1, and that claim 1 is in condition for allowance. Since claims 2-6 depend from claim 1, Applicants contend that claims 2-6 are likewise in condition for allowance.

In addition with respect to claim 3, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 4-6 does not teach the feature: "for each item of control flow code, before identifying any new procedure in accordance with step (b) of the method, checking for compliance between one or more predetermined rules for the software and the software should step (b) and following steps of the method take place".

Also with respect to claim 3, Applicants respectfully contend that Bretemitz Prior Art,

col. 2, line 56 - col. 3, line 14 does not teach the feature: "for that item of control flow code, continuing with step (b) and the following steps of the method only in the event of such compliance".

In addition with respect to claim 4, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 47-49 does not teach the feature: "wherein step (e) comprises the application of a cost-analysis algorithm based on predetermined rules about the length of the software".

In addition with respect to claim 5, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 44-57 does not teach the feature: "optimising the or each new procedure for which a call parameter is a constant by propagating that constant through the new procedure".

In addition with respect to claim 6, Applicants respectfully contend that Breternitz Prior Art, col. 3, line 2 and/or col. 3, lines 10-16 and/or col. 3, line 5 does not teach the feature: "analysing a call statement, calling parameters and an associated new procedure to determine if they are compliant with predetermined in-lining rules and, if they are so compliant, replacing said call statement by a copy of the new procedure".

Claim 7

The Examiner rejected claim 7 under 35 U.S.C. §102(e) as allegedly being anticipated by Breternitz Prior Art.

Applicants respectfully contend that Breternitz Prior Art does not anticipate claim 7, because Breternitz Prior Art does not teach each and every feature of claim 7.

As a first example of why Breternitz Prior Art does not anticipate claim 7, Breternitz Prior Art does not teach the feature: "computer software that includes one or more call statements and a procedure which is callable by the or each call statement".

The Examiner argues that Breternitz, col. 1, line 26 and col. 2, line 4 teaches the preceding feature of claim 7. In response, Applicants respectfully contend that Breternitz, col. 1,

line 26 and col. 2, line 4 does not disclose a call statement and does not disclose a procedure which is callable by the or each call statement, and therefore does not teach the preceding feature of claim 7.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all regarding "Bretemitz does not anticipates a call statement and does not disclose call parameter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been defined as calling a procedure function..... Bretemitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Bretemitz, Col. 2, Line 40- 60, which has been used to make the rejection, and the rejection still stands".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Bretemitz Invention is specific to FIG. 14 of Bretemitz Invention and does not apply to the rejection under Bretemitz Prior Art. In particular, col. 1, lines 25-40 of Bretemitz Prior Art makes it clear that the control flow graph of FIG. 1 of Bretemitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Bretemitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Bretemitz Prior Art are not disclosed as representing subroutines or function calls in Bretemitz Prior Art. The Examiner cannot combine Bretemitz Invention with Bretemitz Prior Art under 35 U.S.C. §102.

As a second example of why Bretemitz Prior Art does not anticipate claim 7, Bretemitz Prior Art does not teach the feature: "if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1

teaches the preceding feature of claim 7.

In response, Applicants respectfully contend that Bretcrnitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 does not identify a new procedure as required by claim 7, and most certainly does not identify a new procedure "if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables". Applicants maintain that Bretcrnitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 is totally irrelevant to the preceding feature of claim 7.

As a third example of why Bretcrnitz Prior Art does not anticipate claim 7, Bretcrnitz Prior Art does not teach the feature: "recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition".

The Examiner argues that Bretcrnitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 teaches the preceding feature of claim 7.

In response, Applicants respectfully contend that Bretcrnitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 does not disclose the preceding feature of claim 7. Bretcrnitz Prior Art, col. 2, lines 14-15 merely discloses storing block execution order in a time sequential manner. Bretcrnitz Prior Art, col. 2, lines 41-45 merely discloses scanning trace data and analyzing pairs of basic blocks that are adjacent to each other in time. Bretcrnitz Prior Art, col. 2, lines 41-45 does not discloses a list, and most certainly not disclose a list of data entries corresponding to the respective new procedures. Applicants maintain that Bretcrnitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 is totally irrelevant to the preceding feature of claim 7.

As a fourth example of why Bretcrnitz Prior Art does not anticipate claim 7, Bretcrnitz Prior Art does not teach the feature: "for each said call statement, scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement".

The Examiner argues that Bretcrnitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60

teaches the preceding feature of claim 7.

In response, Applicants respectfully contend that Bretcmitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the preceding "scanning the entries in said list to determine ..." feature of claim 7. Applicants maintain that Bretcmitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 is totally irrelevant to the preceding feature of claim 7.

As a fifth example of why Bretcmitz Prior Art does not anticipate claim 7, Bretcmitz Prior Art does not teach the feature: "modifying the call statements to call said new procedures".

The Examiner argues that Bretcmitz Prior Art, col. 3, lines 34-37 teaches the preceding feature of claim 7.

In response, Applicants respectfully contend that Bretcmitz Prior Art, col. 2, lines 34-37does not disclose the preceding feature of claim 7. The restructured computer file discussed in Bretcmitz Prior Art, col. 2, lines 34-37 is totally irrelevant to "modifying the call statements to call said new procedures" as recited in claim 7.

Based on the preceding arguments, Applicants respectfully maintain that Bretcmitz Prior Art does not anticipate claim 7, and that claim 7 is in condition for allowance.

Claims 8-13

The Examiner rejected claims 8-13 under 35 U.S.C. §102(e) as allegedly being anticipated by Bretcmitz Prior Art.

Applicants respectfully contend that Bretcmitz Prior Art does not anticipate claim 8, because Bretcmitz Prior Art does not teach each and every feature of claim 8.

As a first example of why Bretcmitz Prior Art does not anticipate claim 8, Bretcmitz Prior Art does not teach the feature: "computer software that includes one or more call statements and a procedure which is callable by the or each call statement and which has two or more code branches and control flow code for directing program flow to the code branches".

The Examiner argues that Bretcmitz, col. 1, line 26 and col. 2, line 4 teaches the

preceding feature of claim 8. In response, Applicants respectfully contend that Bretemitz, col. 1, line 26 and col. 2, line 4 does not disclose a call statement and does not disclose call parameters, and therefore does not teach the preceding feature of claim 8.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all regarding "Bretemitz does not anticipates a call statement and does not disclose call paramcter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been defined as calling a procedure function.... Bretemitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Bretemitz, Col. 2, Line 40- 60 ...".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Bretemitz Invention is specific to FIG. 14 of Bretemitz Invention and does not apply to the rejection under Bretemitz Prior Art. In particular, col. 1, lines 25-40 of Bretemitz Prior Art makes it clear that the control flow graph of FIG. 1 of Bretemitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Bretemitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Bretemitz Prior Art are not disclosed as representing subroutines or function calls in Bretemitz Prior Art. The Examiner cannot combine Bretemitz Invention with Bretemitz Prior Art under 35 U.S.C. §102.

As a second example of why Bretemitz Prior Art does not anticipate claim 8, Bretemitz Prior Art does not teach the feature: "analyzing means for analysing the procedure to identify said control flow code and said code branches".

The Examiner argues that Bretemitz Prior Art, col.. 2, lines 4-6 teaches the preceding feature of claim 8.

In response, Applicants respectfully contend that Bretemitz Prior Art, col.. 1, lines 4-6 does not disclose analysing the procedure to "identify said control flow code and said code branches", as required by claim 8. Instead, Bretemitz Prior Art, col.. 2, lines 4-6 discloses analysing the procedure to "determine efficient rearrangement of computer basic blocks in memory so that software executes in an efficient manner." Applicants maintain that Bretemitz

Prior Art, col. 2, line 4-6 is totally irrelevant to the preceding feature of claim 8.

As a third example of why Bretemitz Prior Art does not anticipate claim 8, Bretemitz Prior Art does not teach the feature: "identifying means for identifying each said code branch a new procedure containing the respective code branch".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 10-13 teaches the preceding feature of claim 8.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, line 10-13, does not identify a new procedure, as required by claim 8. Bretemitz Prior Art, col. 2, line 10-13 merely discusses a trace data file the records time-sequential execution flow of the computer program.. Applicants maintain that Bretemitz Prior Art, col. 2, line 10-13 is totally irrelevant to the preceding feature of claim 8.

As a fourth example of why Bretemitz Prior Art does not anticipate claim 8, Bretemitz Prior Art does not teach the feature: "recording means for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 teaches the preceding feature of claim 8.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 does not disclose the preceding feature of claim 8. Bretemitz Prior Art, col. 2, lines 14-15 merely discloses storing block execution order in a time sequential manner. Bretemitz Prior Art, col. 2, lines 41-45 merely discloses scanning trace data and analyzing pairs of basic blocks that are adjacent to each other in time. Bretemitz Prior Art, col. 2, lines 41-45 does not discloses a list, and most certainly not disclose a list of data entries corresponding to the respective new procedures. Applicants maintain that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 is totally irrelevant to the preceding feature of claim 8.

As a fifth example of why Bretemitz Prior Art does not anticipate claim 8, Bretemitz Prior Art does not teach the feature: "scanning means operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters supplied by the call statement".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 teaches the preceding feature of claim 8.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the preceding "scanning the entries in said list to determine ..." feature of claim 8. Applicants maintain that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 is totally irrelevant to the preceding feature of claim 8.

As a sixth example of why Bretemitz Prior Art does not anticipate claim 8, Bretemitz Prior Art does not teach the feature: "modifying means for modifying the call statement to call the corresponding new procedure".

The Examiner argues that Bretemitz Prior Art, col. 3, lines 34-37 teaches the preceding feature of claim 8.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 34-37does not disclose the preceding feature of claim 8. The restructured computer file discussed in Bretemitz Prior Art, col. 2, lines 34-37 is totally irrelevant to "modifying the call statement to call the corresponding new procedure" as recited in claim 8.

Based on the preceding arguments, Applicants respectfully maintain that Bretemitz Prior Art does not anticipate claim 8, and that claim 8 is in condition for allowance. Since claims 9-13 depend from claim 8, Applicants contend that claims 9-13 are likewise in condition for allowance.

In addition with respect to claim 10 Applicants respectfully contend that Bretemitz Prior Art, col. 2, line 56 - col. 3, line 14 does not teach the feature: "wherein the system comprises checking means which is operable to check for compliance between one or more predetermined

rules for the software and the software should said identifying means identify any new procedure." Breternitz Prior Art, col. 2, line 56 - col. 3, line 14 does not disclose said one or more predetermined rules if the condition "should said identifying means identify any new procedure" is satisfied.

In addition with respect to claim 11 Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 47-49 does not teach the feature: "wherein said checking means is operable for checking compliance with a cost-analysis algorithm based on predetermined rules about the length of the software".

In addition with respect to claim 12 Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 44-57 does not teach the feature: "means for optimising the or each new procedure for which a call parameter is a constant by propagating that constant through the new procedure".

In addition with respect to claim 13 Applicants respectfully contend that Breternitz Prior Art, col. 3, line 2 and/or col. 3, lines 10-16 and/or col. 3, line 5 does not teach the feature: "means for analysing a call statement, calling parameters and an associated new procedure to determine if they are compliant with predetermined related to in-lining and, if they are so compliant, replacing said call statement by a copy of the new procedure".

Claim 14

The Examiner rejected claim 14 under 35 U.S.C. §102(e) as allegedly being anticipated by Breternitz Prior Art.

Applicants respectfully contend that Breternitz Prior Art does not anticipate claim 14, because Breternitz Prior Art does not teach each and every feature of claim 14.

As a first example of why Breternitz Prior Art does not anticipate claim 14, Breternitz Prior Art does not teach the feature: "computer software that includes one or more call statements

and a procedure which is callable by the or each call statement".

The Examiner argues that Bretemitz, col. 1, line 26 and col. 2, line 4 teaches the preceding feature of claim 14. In response, Applicants respectfully contend that Bretemitz, col. 1, line 26 and col. 2, line 4 does not disclose a call statement and does not disclose a procedure which is callable by the or each call statement, and therefore does not teach the preceding feature of claim 14.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all regarding "Bretemitz does not anticipates a call statement and does not disclose call parameter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been defined as calling a procedure function..... Bretemitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Bretemitz, Col. 2, Line 40- 60, which has been used to make the rejection, and the rejection still stands".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Bretemitz Invention is specific to FIG. 14 of Bretemitz Invention and does not apply to the rejection under Bretemitz Prior Art. In particular, col. 1, lines 25-40 of Bretemitz Prior Art makes it clear that the control flow graph of FIG. 1 of Bretemitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Bretemitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Bretemitz Prior Art are not disclosed as representing subroutines or function calls in Bretemitz Prior Art. The Examiner cannot combine Bretemitz Invention with Bretemitz Prior Art under 35 U.S.C. §102.

As a second example of why Bretemitz Prior Art does not anticipate claim 14, Bretemitz Prior Art does not teach the feature: "means for traversing the control flow graph to consider each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the

nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered".

The Examiner argues that Breternitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 teaches the preceding feature of claim 14.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 does not identify a new procedure as required by claim 14, and most certainly does not identify a new procedure "if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables". Applicants maintain that Breternitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 is totally irrelevant to the preceding feature of claim 14.

As a third example of why Breternitz Prior Art does not anticipate claim 14, Breternitz Prior Art does not teach the feature: "means for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition".

The Examiner argues that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 teaches the preceding feature of claim 14.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 does not disclose the preceding feature of claim 14. Breternitz Prior Art, col. 2, lines 14-15 merely discloses storing block execution order in a time sequential manner. Breternitz Prior Art, col. 2, lines 41-45 merely discloses scanning trace data and analyzing pairs of basic blocks that are adjacent to each other in time. Breternitz Prior Art, col. 2, lines 41-45 does not discloses a list, and most certainly not disclose a list of data entries corresponding to the respective new procedure. Applicants maintain that Breternitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 is totally irrelevant to the preceding feature of claim 14.

As a fourth example of why Breternitz Prior Art does not anticipate claim 14, Breternitz

Prior Art does not teach the feature: "means for scanning the entries in said list to determine for each call statement, an entry for which there is correspondence between said branch condition and call parameters supplied by the call statement".

The Examiner argues that Breternitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 teaches the preceding feature of claim 14.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the preceding "scanning the entries in said list to determine ..." feature of claim 14. Applicants maintain that Breternitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 is totally irrelevant to the preceding feature of claim 14.

As a fifth example of why Breternitz Prior Art does not anticipate claim 14, Breternitz Prior Art does not teach the feature: "means for modifying the call statements to call said new procedures".

The Examiner argues that Breternitz Prior Art, col. 3, lines 34-37 teaches the preceding feature of claim 14.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 34-37does not disclose the preceding feature of claim 14. The restructured computer file discussed in Breternitz Prior Art, col. 2, lines 34-37 is totally irrelevant to "modifying the call statements to call said new procedures" as recited in claim 14.

Based on the preceding arguments, Applicants respectfully maintain that Breternitz Prior Art does not anticipate claim 14, and that claim 14 is in condition for allowance.

Claim 15

The Examiner rejected claim 15 under 35 U.S.C. §102(e) as allegedly being anticipated by Breternitz Prior Art.

Applicants respectfully contend that Breternitz Prior Art does not anticipate claim 15, because Breternitz Prior Art does not teach each and every feature of claim 15.

As a first example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "computer software that includes one or more call statements and a procedure which is callable by the or each call statement and which has two or more code branches and control flow code for directing program flow to the code branches".

The Examiner argues that Bretemitz, col. 1, line 26 and col. 2, line 4 teaches the preceding feature of claim 15. In response, Applicants respectfully contend that Bretemitz, col. 1, line 26 and col. 2, line 4 does not disclose a call statement and does not disclose call parameters, and therefore does not teach the preceding feature of claim 15.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all regarding "Bretemitz does not anticipates a call statement and does not disclose call parameter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been defined as calling a procedure function.... Bretemitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Bretemitz, Col. 2, Line 40- 60 ...".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Bretemitz Invention is specific to FIG. 14 of Bretemitz Invention and does not apply to the rejection under Bretemitz Prior Art. In particular, col. 1, lines 25-40 of Bretemitz Prior Art makes it clear that the control flow graph of FIG. 1 of Bretemitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Bretemitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Bretemitz Prior Art are not disclosed as representing subroutines or function calls in Bretemitz Prior Art. The Examiner cannot combine Bretemitz Invention with Bretemitz Prior Art under 35 U.S.C. §102.

As a second example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "a first computer code portion for analysing the procedure to identify said control flow code and said code branches".

The Examiner argues that Bretemitz Prior Art, col.. 2, lines 4-6 teaches the preceding feature of claim 15.

In response, Applicants respectfully contend that Bretemitz Prior Art, col.. 1, lines 4-6 does not disclose analysing the procedure to "identify said control flow code and said code branches", as required by claim 15. Instead, Bretemitz Prior Art, col.. 2, lines 4-6 discloses analysing the procedure to "determine efficient rearrangement of computer basic blocks in memory so that software executes in an efficient manner." Applicants maintain that Bretemitz Prior Art, col. 2, line 4-6 is totally irrelevant to the preceding feature of claim 15.

As a third example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "a second computer code portion for identifying for each said code branch a new procedure containing the respective code branch".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 10-13 teaches the preceding feature of claim 15.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, line 10-13, does not identify a new procedure, as required by claim 15. Bretemitz Prior Art, col. 2, line 10-13 merely discusses a trace data file the records time-sequential execution flow of the computer program.. Applicants maintain that Bretemitz Prior Art, col. 2, line 10-13 is totally irrelevant to the preceding feature of claim 15.

As a fourth example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "a third computer code portion for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 teaches the preceding feature of claim 15.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 does not disclose the preceding feature of claim 15. Bretemitz Prior Art, col. 2, lines 14-15 merely discloses storing block execution order in a time sequential manner. Bretemitz Prior Art, col. 2, lines 41-45 merely discloses scanning trace data and analyzing pairs

of basic blocks that are adjacent to each other in time. Bretemitz Prior Art, col. 2, lines 41-45 does not discloses a list, and most certainly not disclose a list of data entries corresponding to the respective new procedures. Applicants maintain that Bretemitz Prior Art, col. 2, lines 14-15 and col. 2, lines 41-45 is totally irrelevant to the preceding feature of claim 15.

As a fifth example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "a fourth computer code portion operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters supplied by the call statement".

The Examiner argues that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 teaches the preceding feature of claim 15.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 does not disclose the preceding "scanning the entries in said list to determine ..." feature of claim 15. Applicants maintain that Bretemitz Prior Art, col. 2, lines 40-51 and col. 2, lines 56-60 is totally irrelevant to the preceding feature of claim 15.

As a sixth example of why Bretemitz Prior Art does not anticipate claim 15, Bretemitz Prior Art does not teach the feature: "a fifth computer code portion for modifying the call statement to call the corresponding new procedure".

The Examiner argues that Bretemitz Prior Art, col. 3, lines 34-37 teaches the preceding feature of claim 15.

In response, Applicants respectfully contend that Bretemitz Prior Art, col. 2, lines 34-37does not disclose the preceding feature of claim 15. The restructured computer file discussed in Bretemitz Prior Art, col. 2, lines 34-37 is totally irrelevant to "modifying the call statement to call the corresponding new procedure" as recited in claim 15.

Based on the preceding arguments, Applicants respectfully maintain that Bretemitz Prior Art does not anticipate claim 15, and that claim 15 is in condition for allowance.

Claims 16-17

The Examiner rejected claims 16-17 under 35 U.S.C. §102(e) as allegedly being anticipated by the combination of Bretemitz Prior Art and Bretemitz Invention (col.. 6, lines 38-39).

Since the Examiner cannot reject claims based on combining references under 35 U.S.C. §102(e), Applicants respectfully contend that the rejection of claims 16-17 is improper and should be withdrawn.

In addition, since claims 16-17 depend from claim 15 which Applicants argued *supra* to not be anticipated by Bretemitz Prior Art, Applicants contend that claims 16-17 are likewise in condition for allowance.

Claim 18

The Examiner rejected claim 18 under 35 U.S.C. §102(e) as allegedly being anticipated by Bretemitz Prior Art.

Applicants respectfully contend that Bretemitz Prior Art does not anticipate claim 18, because Bretemitz Prior Art does not teach each and every feature of claim 18.

As a first example of why Bretemitz Prior Art does not anticipate claim 18, Bretemitz Prior Art does not teach the feature: "computer software that includes one or more call statements and a procedure which is callable by the or each call statement".

The Examiner argues that Bretemitz, col. 1, line 26 and col. 2, line 4 teaches the preceding feature of claim 18. In response, Applicants respectfully contend that Bretemitz, col. 1, line 26 and col. 2, line 4 does not disclose a call statement and does not disclose a procedure which is callable by the or each call statement, and therefore does not teach the preceding feature of claim 18.

In "Response to Arguments", the Examiner argues: "Applicants' arguments over all regarding "Bretemitz does not anticipates a call statement and does not disclose call parameter" (Page 12 -15) is disagreed with. Based on applicants' specification, "call statement" has been

defined as calling a procedure function..... Breternitz anticipates "call statement" [function calls, Col. 16, Line 64- Col. 17, Lines 11], and "call parameters" [1310, Fig. 31], can be used to explain the descriptions in Breternitz, Col. 2, Line 40- 60, which has been used to make the rejection, and the rejection still stands".

In response, Applicant notes that the Examiner's reference to disclosure of call statements in Breternitz Invention is specific to FIG. 14 of Breternitz Invention and does not apply to the rejection under Breternitz Prior Art. In particular, col. 1, lines 25-40 of Breternitz Prior Art makes it clear that the control flow graph of FIG. 1 of Breternitz Prior Art shows basic blocks "a" - "j" which are computer instructions or segments of code. Although the basic blocks of FIG. 14 of Breternitz Invention may represent subroutines or function calls, the basic blocks of FIG. 1 of Breternitz Prior Art are not disclosed as representing subroutines or function calls in Breternitz Prior Art. The Examiner cannot combine Breternitz Invention with Breternitz Prior Art under 35 U.S.C. §102.

As a second example of why Breternitz Prior Art does not anticipate claim 18, Breternitz Prior Art does not teach the feature: "a second code portion for traversing the control flow graph to consider each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered".

The Examiner argues that Breternitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 teaches the preceding feature of claim 18.

In response, Applicants respectfully contend that Breternitz Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 does not identify a new procedure as required by claim 18, and most certainly does not identify a new procedure "if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be

represented as a function only of formal parameters and global variables". Applicants maintain

[illegible] Prior Art, col. 2, lines 10-12, 4-6, 55-60 and Fig. 1 is totally irrelevant to the